LANGLEY GRANT
IN-CAT. 24-CR
270028
20 P

ANNUAL REPORT

FOR

NASA GRANT - NAG-1-431

DETERMINATION OF THE ULTRASONIC PROPERTIES OF COMPOSITES

PERIOD: NOVEMBER 1, 1985 - OCTOBER 31, 1986

(NASA-CR-180974) DETERMINATION OF THE ULTRASONIC PROPERTIES OF COMPOSITES Annual Report, 1 Nov. 1985 - 31 Oct. 1986 (Christopher Newport Coll.) 20 p

N90-70572

Unclas
00/24 0270028

PRINCIPAL INVESTIGATORS:

B.T. SMITH
A.M. BUONCRISTIANI

DEPARTMENT OF PHYSICS
CHRISTOPHER NEWPORT COLLEGE
NEWPORT NEWS, VIRGINIA 23606

NASA TECHNICAL OFFICER:

J.S. HEYMAN, HEAD, MCIS

NASA/LANGLEY RESEARCH CENTER
MAIL STOP 231
HAMPTON, VIRGINIA 23665

INTRODUCTION

This end-of-year report summarizes the progress made on NASA grant NAG-1-141 (The Determination of the Ultrasonic Properties of Composites) and covers the period from November 1, 1985 to October 31, 1986. The work reviewed here includes the impact damage assessment done in collaboration with Sam Freeman of Lockheed-Georgia Company and Jim Miller of Washington University. This work has been submitted to Materials Evaluation in the form of a joint paper. The manuscript is included at the end of this report. There has also been progress in the investigation of the temperature dependence of the ultrasonic acoustical properties of composites. The temperature dependence of the ultrasonic wave group velocity of lexan (polycarbonate) graphite fiber composites has been determined and will be reviewed in this report.

I Temperature Dependence of the Acoustical Properties of Composites

A composite of great interest for aerospace applications is a lamination of graphite fiber arranged in different directions and reinforced with a polymer matrix. The acoustic properties of the composite depend on the acoustic properties of the matrix, fiber and matrix-fiber interface and the direction of propagation in the composite. The matrix is a viscoelastic solid with a low acoustic velocity and high acoustic attenuation. The fiber is a polycrystalline solid with a very high acoustic velocity and low attenuation when compared to the matrix. The acoustic properties of the matrix-fiber interface are unknown. A new experimental technique (1) was introduced in 1984 to take advantage of the unique properties of this composite system to deduce not only material properties but an improved understanding of the acoustic wave propagation in composites.

The interaction of ultrasonic waves and composite materials includes contribution from all the constituent materials. For waves propagating perpendicular to the fiber direction, one expects the velocity to be increased over the velocity of the matrix as the result of the reinforcing fibers. Also, the attenuation in a defect-free composite is expected to result from absorption by the matrix, scattering from the fibers and an absorption by the matrix-fiber interface material. The matrix properties may be seperated from the fiber properties in various ways. One technique involves measuring the acoustic properties as a function of frequency as done by Williams, Lee, and Nabye-Hasheme (2). Our technique takes advantage of the viscoelastic character of the matrix and the matrix-fiber interface. For a viscoelastic solid, as its temperature changes, dramatic changes occur in its attenuation caused by absorption. In contrast, the scattering component should remain approximately constant as the temperature changes. Also, the velocity varies with temperature due to stiffness changes in the viscoelastic solid. By measuring the temperature dependence of the attentuation and velocity of a neat resin and a composite, the wave propagation in the composite can be modeled. The early report sited above (1) assumed a two component model for the attenuation and was able to predict volume fraction of matrix and scattering contribution due to the fibers for a graphite fiber/polysulfone resin composite. The work rviewed in this report is still preliminary and a model to describe the overall temperature response of these lexan samples has not been developed. A description of the experimental procedure will be followed by a review of the data and analysis.

The experimental technique is similar to that reported previously (1). The experimental arrangement has been upgraded to a well insulated liquid bath of ethylene glycol (70% solution) with an immersion heater and an immersion cooler (Neslab 100-CC). This new arrangement allows temperature variations in the range of -50 to +100 centigrade. The specimen under study is immersed, as well as, the transducer in the liquid bath. The transducer used in the experiment reported here is a highly damped broadband transducer with a nominal center frequency of 5 MHz. The transducer is excited with an high voltage (~200 Volt) impulse. The samples that were examined were manufactured with the same cure cycle and included a 100% lexan, 28% lexan, and 51% lexan composite sample. The samples are uniaxial and have 16 plys.

The experimental technique involves digitizing the backscattered wave so as to include the front surface reflection and any internal reflections from the sample. This procedure is performed as as function of temperature. The sampling frequency of the digitizer is 100 MHz and 100 repetitions are averaged to improve the signal to noise. An example of an acquired wave is shown in figure 1 and corresponds to a sample termperature of -7C. The temperature was measured at the sample with a chromel-alumel thermocouple referenced to an electronic zero point. The internal echoes are then fit using the front surface reflection as the input function in a least squares fitting routine to solve for the reflection coefficient, the velocity and the attentuation. The calculations are performed relative to the front surface reflection eliminating any effects of the ultrasonic transmission properties of the liquid that change with temperature. This procedure is performed for both neat resin and composites of various resin content.

In this series of experiments is was difficult to resolve more than the first return from the back surface of the sample. This was due to efficient coupling of the wave into the liquid at the back surface. The calculations of the acoustic parameters of reflection and attenuation did not exhibit a continuous transistion with temperature and will not be included in this preliminary report on this work. The discontinuous nature of the reflection and attenuation coefficients is due to the lack of sufficient data to allow the least square fitting routine to converge(two internal echoes would improve the data reduction algorithm). The calculation of velocity is not as sensitive to convergence and is included in this report. The fitting function used is

$$\frac{-R(1-R^2)e^{-2\alpha d+i\omega 2d/c}}{R}$$

where R is the reflecion coefficient, d is the sample thickness, c is the wave velocity, α is the attenuation and ω is the angular frequency. This term is convolved with the front surface reflected wave and solved in a least square fitting routine for the variables R, c and α. A listing of the fitting program is included in the appendix. Shown in figure 2 is the data of the first internal echo and the fit to the data for a temperature of -7 centigrade. The velocity result for the three samples measured is shown in figure 3. The variation of velocity with temperature is typical of viscoelastic materials.

The results for the attenuation and reflection coefficients were not included. The experiment is in the process of being changed to detect more internal echoes to improve the convergence of the fitting routine. This will be done by bonding another material to the back of the sample to include an air gap. This will permit total reflection of the wave and allow detection of at least two internal echoes.

This report is meant as a review of progress and work will continue with this set of samples. When sufficient data is obtained, modeling efforts will be undertaken to predict fiber volume fraction based on this experimental technique.

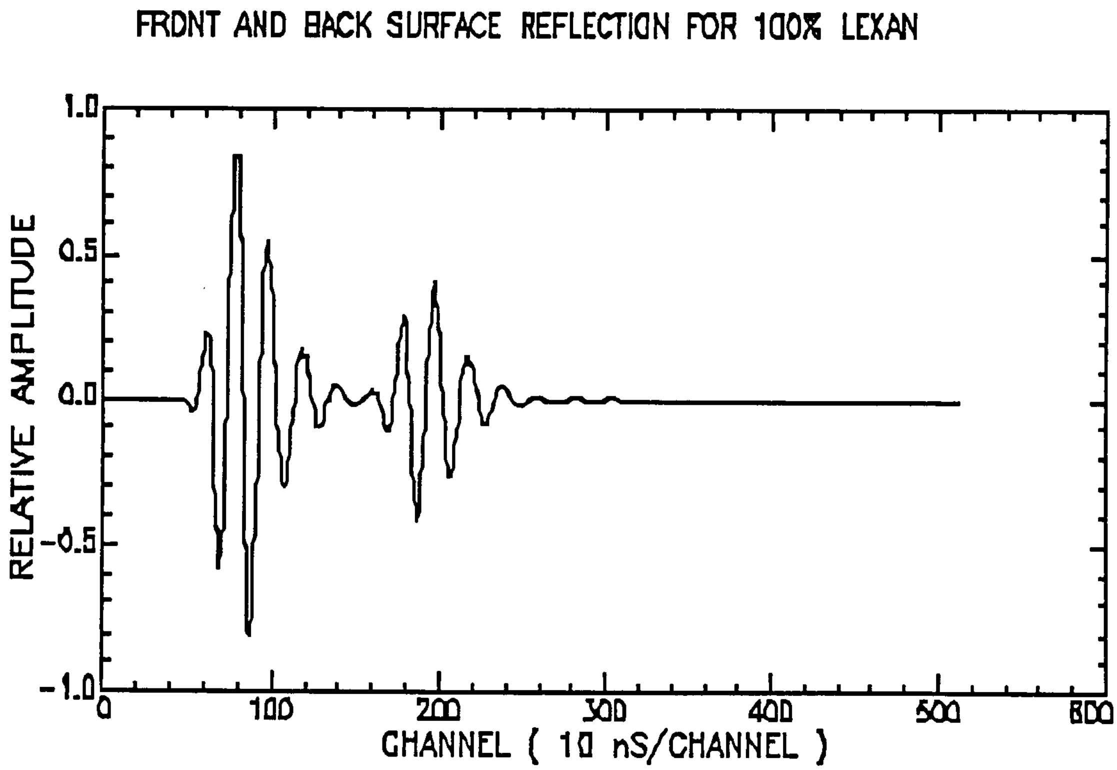
II Impact Damage Assessment in Thin Graphite-Epoxy Composites

The investigation of impact generated delaminations in thin composites of graphite-epoxy has been very successful. In collaboration with Sam Freeman of the Lockheed-Georgia Company and Jim Miller and Earl Blodgett of Washington University, quantitative ultrasonic measurements have been completed. That part of the work completed at the Langley Research Center has made it posssible to detect and measure damage through the entire volume of the composite. The details are included as part of a joint paper which has been submitted to the journal Materials Evaluation. The manuscript follows the work incluced in section one of this report.

REFERENCES

1. B.T. Smith and W.P. Winfree, Proceedings of 1984 IEEE Ultrasonics Symposium, November 14-16, 1984.

2. J.H. Williams, Jr., S.S. Lee, and H. Nayeb-Hashemi, Journal of Nondestructive Evaluation, **1**, 191-199 (1980).



FRONT AND BACK SURFACE REFLECTION FOR 100% LEXAN
RELATIVE AMPLITUDE
1.0
0.5
0.0
-0.5
-1.0
0
100
200
300
400
500
600
CHANNEL (10 nS/CHANNEL)

Figure 1

INTERNAL REFLECTION AND FIT FOR 100% LEXAN

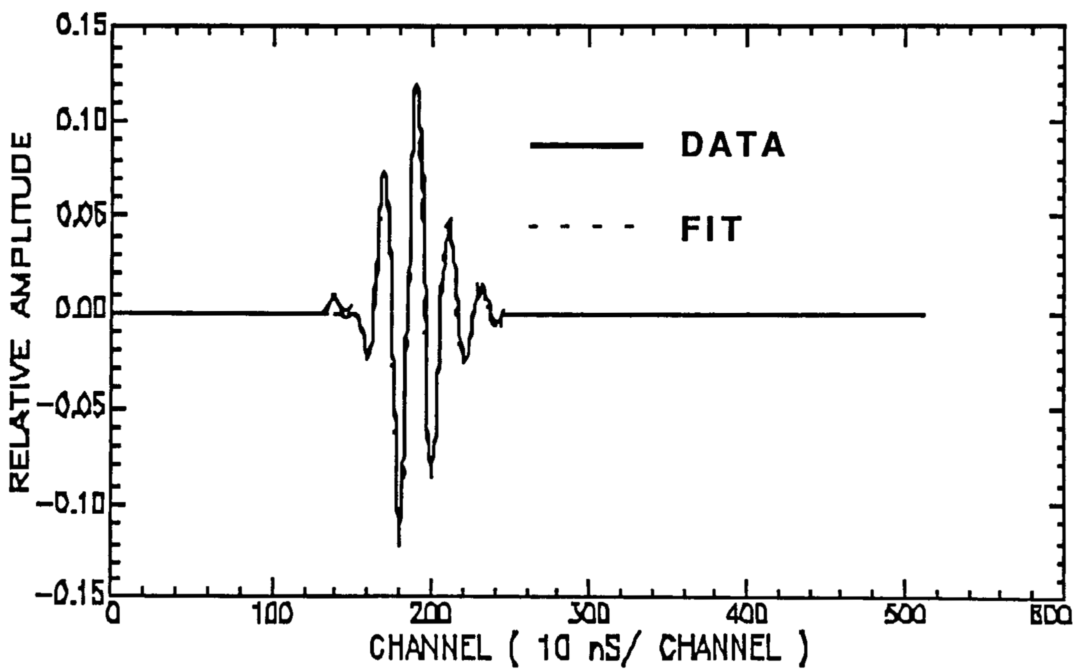


Figure 2

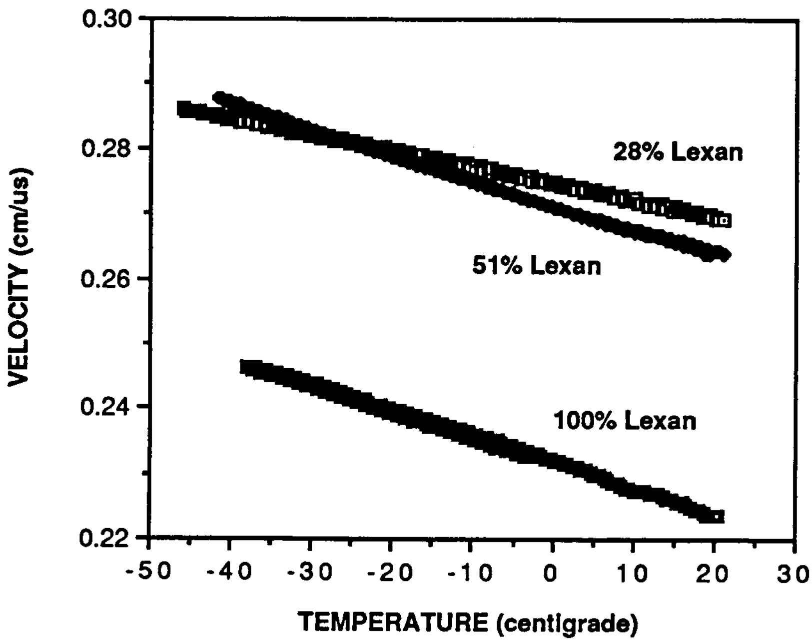

0.30
0.28
0.26
0.24
0.22
VELOCITY (cm/us)
-50
-40
-30
-20
-10
0
10
20
30
TEMPERATURE (centigrade)
28% Lexan
51% Lexan
100% Lexan

Figure 3

APPENDIX

This program was written in Fortran to run on a VAX 11/750 computer

```
C WRITTEN BY W.P. WINFREE
C MODIFIED  BY B.T. SMITH
C does a least sqaures fit of the data against the theoretical
C prediction for the shape of the data for a single acoustic layer.
C
C does the fitting for a series of data set then stores the results in
C a file on disc
C
C parameters: A(1) - reflection coefficient
C
C                 A(2) - velocity in units of cm/usec
C
C                 A(3) - linear attenuation coefficient
C
C
C definitions: DATA - NPTS point array of measured time responce of layer
C
C          REFL - NPTS point array with a reflection of the wave without
C                    the layer behind it
C
C          FIT  - NPTS point array with the fit calculated from A(1) & A(2)
C
C          DERF - 4*NPTS point array with the derivative of the fitting
C                 function with respect to each of the parmameters
C
C          ICH1 - first channel of the fit
C
C          ICH2 - last channel of the fit
C
C         NTERMS- the number of parameters in the fit
C

      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
C INCLUDE common for the curfitting
      DIMENSION A(3),XPLOT(2048)
      REAL*4 DATA,XPLOT
      DIMENSION AC(8)
      CHARACTER*32 NAME
      CHARACTER*1 PLOT_OP,TRY,COM
      CHARACTER*32 RESULTS
      CHARACTER*10 DSD,DST
      INTEGER*4 NPTS
      COM=','

      weight=1.0
      WRITE(6,101)
101   FORMAT('$ DATA SET TO STORE RESULTS IN: ')
      READ(5,4)RESULTS
      OPEN(UNIT=21,FILE=RESULTS,STATUS='NEW')
3     format (a,i3)
4     format (a)
      CALL DATE(DSD)
```

```
        CALL TIME(DST)
        WRITE(6,1)DSD,DST
        WRITE(21,1)DSD,DST
1       FORMAT(' ',5X,' RESULTS OF FIT3'/5X,A,4X,A)


C find the data sets to be fit
        WRITE(6,5)
5       FORMAT('$ NAME OF THE DATA SET TO BE FIT: ')
        ACCEPT '(A)',NAME
        WRITE(6,6)
 6      FORMAT('$ FIRST AND LAST DATA SET NUMBER: ')
        READ(5,*)NUM_DS_F,NUM_DS_L

        WRITE(6,12)
12      FORMAT('$ PLOTTING (Y OR N): ')
        READ(5,3)PLOT_OP

C get the first channel and last channel for reflection
        CALL GETDAT(NAME,NUM_DS_F,AC,DATA,NO_BYTES)
        DO I=1,NO_BYTES

        XPLOT(I)=I
        ENDDO

        IF (ICHAR(PLOT_OP).EQ.ICHAR('Y')) THEN


        CALL CMDLIN('DEV VW SIZE 7,13 PLOT')

        CALL GENPLT(XPLOT,DATA,NO_BYTES)
        END IF

        XMAX=DATA(1)
        DO I=2,NO_BYTES

        XMAX=AMAX1(DATA(I),XMAX)
        ENDDO
        DO I=1,NO_BYTES

        IF (XMAX .EQ. DATA(I))GO TO 103
        ENDDO
 103    IMAX=I

7       TYPE*,'# CHANNEL EITHER SIDE OF Refl. pulse (IMAX +/- DELTA I EVEN # '
        READ (5,*)DELTAI
        ICHAN1=IMAX-DELTAI
        IF(ICHAN1 .LE. 0)ICHAN1=1
        ICHAN2=IMAX+DELTAI
        WRITE(6,8) ICHAN1,ICHAN2
        WRITE(21,8) ICHAN1,ICHAN2
8       FORMAT(' FIRST CHANNEL: ',I4,' LAST CHANNEL: ',I4)
C input the first and last channel of the fit
!       WRITE(6,9)
9       FORMAT('$ FIRST & LAST CHANNEL OF THE FIT ')
!       READ(5,*)ICH1,ICH2
        !,ich3,ich4
!       ICH1=ICHAN2
!       ICH2=NO_BYTES-25
```

```
      TYPE*,'# CHANNEL EITHER SIDE OF INTERNAL pulse (IMAX +/- DELTA '
      READ (5,*)DELTAI1

      XMAX=DATA(IMAX+DELTAI+1)
      DO I=IMAX+DELTAI+1,NO_BYTES

      XMAX=AMAX1(DATA(I),XMAX)
      ENDDO
      DO I=IMAX+DELTAI+1,NO_BYTES

      IF (XMAX .EQ. DATA(I))GO TO 105
      ENDDO
 105  IMAX=I

      ICH1=IMAX-DELTAI1
      ICH2=IMAX+DELTAI1



      WRITE(6,10)ICH1,ICH2
      !,ich3,ich4
      WRITE(21,10)ICH1,ICH2
      !,ich3,ich4
10    FORMAT(' FIRST AND LAST CHANNELS OF THE FIT: ',4I6)

      WRITE(6,13)
13    FORMAT('$ THICKNESS OF THE LAYER (INCHES): ')
      READ(5,*)THICKNESS
      WRITE(21,*)NAME


 999  CONTINUE

C input the first quess of the parameters & find the initial chisqr



      WRITE(6,11)
11    FORMAT('$ A(3) : R,vel(cm/usec),atten. a(3)')
      READ(5,*)A

      WRITE(6,131)THICKNESS
      WRITE(21,131)THICKNESS
131   FORMAT(' THICKNESS: ',F8.4,' INCHES')
C
C start of the loop that does the fitting
C

      I_STEP=1
      IF(NUM_DS_F.GT.NUM_DS_L) I_STEP=-1
      DO NUM=NUM_DS_F,NUM_DS_L,I_STEP
      ifirst=0
      FLAMDA=.0001
C input the data set to be fit
      CALL GETDAT(NAME,NUM,AC,DATA,NO_BYTES)
      TIME_BASE=AC(3)*1.0E-3
      TEMP=AC(2)
```

```
C FIND THE MAX AND CHANNEL RANGE

      XMAX=DATA(1)
      DO I=2,NO_BYTES

      XMAX=AMAX1(DATA(I),XMAX)
      ENDDO
      DO I=1,NO_BYTES

      IF (XMAX .EQ. DATA(I))GO TO 104
      ENDDO
 104  IMAX=I

      ICHAN1=IMAX-DELTAI
      IF(ICHAN1 .LE. 0)ICHAN1=1
      ICHAN2=IMAX+DELTAI
!     ICH1=ICHAN2
!     ICH2=NO_BYTES-25

      XMAX=DATA(IMAX+DELTAI+1)
      DO I=IMAX+DELTAI+1,NO_BYTES

      XMAX=AMAX1(DATA(I),XMAX)
      ENDDO
      DO I=IMAX+DELTAI+1,NO_BYTES

      IF (XMAX .EQ. DATA(I))GO TO 106
      ENDDO
 106  IMAX=I

      ICH1=IMAX-DELTAI1
      ICH2=IMAX+DELTAI1

C zero out the data from before the pulse
C find base line
      SUM_D=0.0
      DO I=1,ICHAN1
      SUM_D=SUM_D+DATA(I)
      ENDDO
      BL_D=SUM_D/ICHAN1
C subtract the base line from the data and reflection
      DO I=1,NO_BYTES
      DATA(I)=DATA(I)-BL_D
      REFL(I)=DATA(I)
      ENDDO
C zero out before the first channal
      DO I=ICHAN1,ICH1
      DATA(I)=DATA(I)*EXP(-(I-ICH1)**2/20.)
      ENDDO
      DO I=1,ICHAN1
      DATA(I)=0
      ENDDO

C zero out after the last channel

      DO I=ICH2,NO_BYTES
      DATA(I)=DATA(I)*EXP(-(I-ICH2)**2/20.)
!     DATA(I)=0
```

```
      ENDDO

C zero out before the first channel

      DO I=1,ICHAN1
      REFL(I)=REFL(I)*EXP(-(I-ICHAN1)**2/20.)
!     REFL(I)=0.0
      ENDDO

C zero out after the last channel

      DO I=ICHAN2,ICH1
      REFL(I)=REFL(I)*EXP(-(I-ICHAN2)**2/20.)
      ENDDO
      DO I=ICH1,NO_BYTES
      REFL(I)=0.0
      ENDDO

C zero out the middle of the pulse echo between 1st and 3rd

!     do i=ich3,ich4
!     data(i)=0.0
!     end do


      CALL FIND_FIT(A)
      NPTS=ICH2+1-ICH1
      IF (ICHAR(PLOT_OP).EQ.ICHAR('Y')) THEN

      CALL CMDLIN('XREGION ICH1,ICH2')

      CALL CMDLIN('DEV VW SIZE 7,13 LTYPE 1 PLOT RET')

      CALL GENPLT(XPLOT,DATA,NO_BYTES)


      CALL CMDLIN('LTYPE 2 OVERLAY RET')

      CALL GENPLT(XPLOT,FIT,NO_BYTES)

      TYPE *,'(C)ONTINUE OR (R)ETRY INITIAL GUESS OF PARAMETERS'
      ACCEPT '(A)',TRY
      IF (TRY .EQ. 'R' .OR. TRY .EQ. 'r')go to 999

      END IF



      CALL FIND_CHISQR(CHISQR2)
      CHISQR1=CHISQR2*2
      WRITE(6,15)A,CHISQR2
15    FORMAT(2X,'R:',G11.5,' C:',G11.5,' CA:',G11.5,' LA:',G11.5,
     * ' CS:',G11.5)
      NUMBER=0
      DO WHILE ((1.-CHISQR2/CHISQR1).GT..00001.AND.NUMBER.LT.20)
      NUMBER=NUMBER+1
      CHISQR1=CHISQR2
      CALL CURFIT(A,CHISQR2)
      WRITE(6,15)A,CHISQR2
```

```
      ENDDO
      NPTS=ICH2+1-ICH1
      XMN=0.0
      XMX=1.0
      VELOCITY=A(2)
      WRITE(6,16)VELOCITY
16    FORMAT(' VELOCITY (cm/us): ',F10.4)
      WRITE(21,17)NUM,COM,TEMP,COM,VELOCITY,COM,A(1),COM,A(3),COM,CHISQR2
17    FORMAT(' ',I3,A,F8.3,A,F10.5,3(A,F11.5))


      IF (ICHAR(PLOT_OP).EQ.ICHAR('Y')) THEN


      CALL CMDLIN('DEV VW LTYPE 1 SIZE 7,13 PLOT RET')

      CALL GENPLT(XPLOT,DATA,NO_BYTES)



      CALL CMDLIN('LTYPE 2 OVERLAY RET')

      CALL GENPLT(XPLOT,FIT,NO_BYTES)

      END IF

      ENDDO
      CALL GENOFF
      STOP
      END
C______________________________________________________________________
      SUBROUTINE SPECTRA(D1,A)
      DIMENSION D1(2048),A(3),XPLOT(2048)
      COMPLEX CD1(2048),CEXP,CMPLX,CONJG,P,R1,R2,R3,p2,p4,cd2(2048),r6,p6
      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
C set the data in a complex array
      N=NO_BYTES

      DO I=1,NO_BYTES
      XPLOT(I)=I
      ENDDO

      ifirst=ifirst+1
      if(ifirst.gt.1) go to 200
      CALL CARRAY(REFL,CD1,N)
C find the Fourier transform of the data

      INV=-1
      CALL SFFT(CD1,N,INV)
C calculate the change in phase angle as a function of frequency
200   continue
      d=thickness*2.54
      N2=N/2+1
      FREQS=1.0/(FLOAT(N)*TIME_BASE)
      FREQ=0.0
      WS=6.283185*FREQS
      W=0.0
C the coefficients for multiplication
C ARE SET FOR LIQUID/SPECIMEN/LIQUID EXPERIMENTAL ARRANGEMENT
```

```
      R1=CMPLX(A(1),0.0)
      R2=CMPLX(A(1)*(1.0-A(1)*A(1)),0.0)
!     R3=CMPLX((A(1)**3)*(1-A(1)*A(1)),0.0)
!     r6=cmplx((a(1)**2)*((1-a(1)*a(1))),0.0)
C zero frequency term
!     CD2(1)=CD1(1)*(-R2-R3)/R1
      CD2(1)=CD1(1)*(-R2)/R1
C phase shift all of the different frequency components
    DO I=2,N2
      W=W+WS
      FREQ=FREQ+FREQS
C if the attenuation gets to large set exponent equal to zero
      ATTEN=A(3)
      IF(ABS(ATTEN).GT.25) THEN
!     TYPE *,'ATTEN= ',ATTEN,'(GREATER THAN 25)'
      P2=CMPLX(0.0,0.0)
      P2=CEXP(P2)
      P4=CMPLX(0.0,0.0)
      P4=CEXP(P4)
      ELSE
      P2=CMPLX(-ATTEN*2*d,-W*2*d/a(2))
      P2=CEXP(P2)
!     P4=CMPLX(-ATTEN*4*d,-W*4*d/a(2))
!     P4=CEXP(P4)
!     p6=cmplx(-atten*6*d,-w*6*d/a(2))
!     p6=cexp(p6)

      END IF
!     CD2(I)=CD1(I)*(-R2*P2-R3*P4)/R1
      CD2(I)=CD1(I)*(-R2*P2)/R1
C if there is a negative frequency component set it equal to the complex
C conjugate of the positive frequency component
      IF(I.NE.N2) THEN
      I2=N+2-I
      CD2(I2)=CONJG(CD2(I))
      END IF
      ENDDO
      INV=1
      CALL SFFT(CD2,N,INV)
      CALL RARRAY(D1,CD2,N)

!     CALL GENPLT(XPLOT,D1,NO_BYTES)

      RETURN
      END
C
C  GETDAT(NAME,NUM,A,DATA,NO_BYTES)
C
C       NAME- FIRST THREE CHARACTERS OF THE NAME OF THE DATA SET
C       NUM - NUMBER OF THE DATA SET
C       A - 8 NUMBER ARRAY WITH THE DATA SET DESCRIPTORS
C       D - NPTS NUMBER ARRAY (DATA)
C
C
      SUBROUTINE GETDAT(NAME,NUM,A,DATA,NO_BYTES)
!     INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
      DIMENSION A(8),TEMP(2),DATA(2048),XR(2048)
      REAL*4 DATA
      CHARACTER*4 FN
```

```
      CHARACTER*32 NAME
      CHARACTER*32 TIME(2)

      ID1=NUM/100
      ID2=NUM/10-ID1*10
      ID3=NUM-10*(ID2+ID1*10)
      C0=ICHAR('0')
      ID1=ID1+C0
      ID2=ID2+C0
      ID3=ID3+C0
      FN(1:1)='.'
      FN(2:2)=CHAR(ID1)
      FN(3:3)=CHAR(ID2)
      FN(4:4)=CHAR(ID3)

      CALL ADDEXT(NAME,FN)
      OPEN(UNIT=10,FILE=NAME,FORM='UNFORMATTED',STATUS='OLD',READONLY)

      READ(10)TIME(1)
!     A(1)=TIME
      READ(10)TEMP(1)
      A(2)=TEMP(1)
      READ(10)STIME,NO_BYTES
      A(3)=STIME
      READ(10) (DATA(I),I=1,NO_BYTES)
      READ(10)TEMP(2)
      A(4)=TEMP(2)
      READ(10)TIME(2)
!     A(5)=TIME

!     DO I=1,NO_BYTES
!
      XR(I)=I
!     ENDDO
!     CALL GENPLT(XR,D,NO_BYTES)

      CLOSE(UNIT=10,DISP='KEEP')

      CALL DELEXT(NAME)
      RETURN

      END


C
C
C general curfitting routine
C
C parameters: A*3 - five independent parameters for the fit
C
C definitions: DATA - 2048 point array of measured time responce of layer
C
C
C          FIT  - 2048 point array with the fit calculated from A(1) & A(2)
C
C          DERF - 3*2048 point array with the derivative of the fitting
C                 function with respect to each of the parmameters
C
C          ICH1 - first channel of the fit
```

```
C
C        ICH2 - last channel of the fit
C
C        NTERMS- the number of parameters in the fit
C
C          WEIGHT- weight of the data calculated from the
C
           standard deviation of the data
C
C
C_______________________________________________________________
      SUBROUTINE CURFIT(A,CHISQR)
      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
      DIMENSION ALPHA(3,3),ARRAY(3,3),BETA(3),B(3),A(3)
      CHISQR1=CHISQR
      CALL FIND_DERF(A)
      DO I=1,NTERMS
      BETA(I)=0.0
      DO J=1,NTERMS
      ALPHA(I,J)=0.0
      ENDDO
      ENDDO
      DO I=ICH1,ICH2
      DO J=1,NTERMS
      BETA(J)=BETA(J)+(DATA(I)-FIT(I))*DERF(J,I)
      DO K=1,J
      ALPHA(J,K)=ALPHA(J,K)+DERF(J,I)*DERF(K,I)
      ENDDO
      ENDDO
      ENDDO
      DO J=1,NTERMS
      DO K=1,J
      ALPHA(K,J)=ALPHA(J,K)
      ENDDO
      ENDDO
C limit the number of warnings
      NUM_WARNING=0
100   DO J=1,NTERMS
      DO K=1,NTERMS
      ARRAY(J,K)=ALPHA(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
      ENDDO
      ARRAY(J,J)=1.+FLAMDA
      ENDDO
      CALL MATINV(ARRAY)
      DO J=1,NTERMS
      B(J)=A(J)
      DO K=1,NTERMS
      B(J)=B(J)+BETA(K)*ARRAY(J,K)/SQRT(ALPHA(J,J)*ALPHA(K,K))
      ENDDO
      ENDDO
      CALL FIND_FIT(B)
      CALL FIND_CHISQR(CHISQR)
      IF(CHISQR.LE.CHISQR1) GO TO 200
C     WRITE(6,8)FLAMDA,B,CHISQR
8     FORMAT(' WARNING: INCREASE IN CHISQR!  FLAMDA:',G12.5/
     * 2X,5G12.6)
      FLAMDA=FLAMDA*10.
      NUM_WARNING=NUM_WARNING+1
      IF(NUM_WARNING.GT.20) GO TO 200
```

```
      GO TO 100
200   DO I=1,NTERMS
      A(I)=B(I)

      If ( Abs( 10.**20 * Alpha ( I, I ) )
    1     .gt. Abs ( Array (I, I) ) ) Then

      Sigmaa(I) = Sqrt ( Abs( Array(I, I) / Alpha (I, I) ) )
      End if

      ENDDO
      FLAMDA=FLAMDA*.1
      RETURN
      END
C_______________________________________________________________
      SUBROUTINE FIND_FIT(A)
      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
      DIMENSION A(3)
      CALL SPECTRA(FIT,A)

      RETURN
      END
C_______________________________________________________________
      SUBROUTINE FIND_DERF(A)
      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
      DIMENSION A(3),B(3),D1(2048)
C find the derivative
      DO J=1,NTERMS
      B(J)=A(J)
      ENDDO
      DO J=1,NTERMS
C find the function for the parameter slight perturbed
      B(J)=A(J)*1.001
      CALL SPECTRA(D1,B)
      B(J)=A(J)
      DO I=1,2048
      DERF(J,I)=(D1(I)-FIT(I))/(B(J)*.001)
      ENDDO
      ENDDO
      RETURN
      END
C_______________________________________________________________
      SUBROUTINE FIND_CHISQR(CHI)
      INCLUDE 'DUA1:[BTSMITH.FOR]fit3.cmn'
      DOUBLE PRECISION SUM,DBLE
      SUM=0.0
      DO I=ICH1,ICH2
      SUM=SUM+DBLE(FIT(I)-DATA(I))**2
      ENDDO
      DEN=FLOAT(ICH2+1-(ICH1+NTERMS))
      CHI=SUM*WEIGHT/DBLE(DEN)
      RETURN
      END
C_______________________________________________________________
      SUBROUTINE MATINV(A)
C finds the inverse of a 3 by 3 matrix
C
     DIMENSION A(3,3),B(3,3),JK(3)
C size of the matrix
```

```
      NO=3
      DO 100 I=1,NO
      DO 50 J=1,NO
50    B(I,J)=0.0
100   B(I,I)=1.0
      DO 200 K=1,NO
      AMAX=0.0
      DO 110 I=K,NO
      DO 110 J=K,NO
      IF(AMAX.GT.ABS(A(I,J)))GO TO 110
101   AMAX= ABS(A(I,J))
      IK=I
      JK(K)=J
110   CONTINUE
      IF(AMAX)120,120,130
120   WRITE(6,1)
1     FORMAT(' MATRIX HAS NO INVERSE')
      RETURN
130   I=IK
      IF(I-K)150,150,140
140   DO 141 J=1,NO
      S=A(I,J)
      A(I,J)=A(K,J)
      A(K,J)=S
      S=B(I,J)
      B(I,J)=B(K,J)
141   B(K,J)=S
150   J=JK(K)
      IF(J-K)170,170,160
160   DO 161 I=1,NO
      S=A(I,J)
      A(I,J)=A(I,K)
161   A(I,K)=S
170   DO 175 I=NO,1,-1
175   B(K,I)=B(K,I)/A(K,K)
      DO 180 I=NO,K,-1
180   A(K,I)=A(K,I)/A(K,K)
      IF(K-NO)190,200,200
190   DO 193 I=K+1,NO
      DO 191 J=1,NO
191   B(I,J)=B(I,J)-B(K,J)*A(I,K)
      DO 192 J=NO,K,-1
192   A(I,J)=A(I,J)-A(K,J)*A(I,K)
193   CONTINUE
200   CONTINUE
      DO 300 I=1,NO
      DO 210 J=NO,1,-1
      IF(J-NO) 201,210,210
201   DO 202 K=NO,J+1,-1
202   B(J,I)=B(J,I)-A(J,K)*B(K,I)
210   B(J,I)=B(J,I)/A(J,J)
300   CONTINUE
      DO 400 K=NO,1,-1
      J=JK(K)
      IF(J-K)310,400,310
310   DO 320 I=1,NO
      S=B(J,I)
      B(J,I)=B(K,I)
320   B(K,I)=S
```

```
400  CONTINUE
     DO 500 I=1,NO
     DO 500 J=1,NO
500  A(I,J)=B(I,J)
     RETURN
     END
```